AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph beginning on page 6, line 3 with the following rewritten paragraph:

Referring to Figure 1A, the ~~The~~ invention is a system and method for enabling Intellectual Property Blocks (IP) Blocks 10 and 20 to be reused at a system level. The present invention represents the IP blocks 10 and 20 as blocks that exchange messages without needing to represent the functionality of the IP blocks 10 and 20. The implementations of these IP blocks 10 and 12 exchanges messages through complex signaling protocols 12 and 22. In conventional systems, interfacing between IP blocks that use different signaling protocols is a tedious and error prone design task. The present invention uses regular expression based protocol descriptions to map the messages onto a signaling protocol. Given the two protocols 12 and 22, the present invention uses an interface generator 118 and builds an interface machine 30 that automatically labels data reference by all protocols. The present invention is also capable of generating the interface 30 even when the data sequencing of the two protocols 12 and 22 differs.

Please replace the paragraph beginning on page 6, line 22 with the following rewritten paragraphs:

Figure 1A is an illustration of IP blocks having different signaling protocols and an interface generated for communication between IP blocks according to one embodiment;

Figure 1B ~~1~~ is an illustration of a computer environment in which the preferred embodiment of the present invention may operate.

Please replace the paragraph beginning on page 8, line 15 with the following rewritten paragraph:

Figure 1B ~~1~~ is an illustration of a computer system in which the preferred embodiment of the present invention resides and operates. The computer system includes a conventional computer 102, such as a SPARC Station 10 that is commercially available from Sun Microsystems, Santa Clara, California or an IBM compatible personal computer that is commercially available from IBM Corp., Armonk, NY, for example. The computer 102 includes a memory module 104, a processor 106, an optional network interface 108, a storage device 110, and an input/output (I/O) unit 112. In addition, an input device 122 and a display unit 124 can be coupled to the computer. The memory module 104 can be conventional random access memory

(RAM) and can include a conventional operating system 114, a data module 116 for storing data and data structure generated by the present invention, and application programs 120, including an interface generator 118 (in which the present invention is stored and executed from in the preferred embodiment) and other simulation programs can be stored. Although the preferred embodiment of the present invention is described with respect to a circuit synthesis tool in a computer aided design (CAD) or electronic design automation (EDA) environment, it will be apparent to persons skilled in the art that the system and method of the present invention can be utilized in many different environments or types of circuit syntheses tools and circuit simulators, e.g., timing simulators, mixed signal simulators, logic simulators, etc., without departing from the scope of the present invention.

Please replace the paragraph beginning on page 10, line 16 with the following rewritten paragraph:

Figure 2 is a flow chart describing the operation of the preferred embodiment of the present invention. As described above, the present invention generates an interface 30 between two protocols 12 and 22 of IP blocks 10 and 20. The present invention receives 202 regular expressions representing the first and second protocols 12 and 22. The following description is based upon generating an interface 30 between two protocols 12 and 22. Additional interfaces can be generated between additional protocols by repeatedly performing the following processes, for example. After receiving 202 the regular expressions, the present invention generates 204 finite automata from the regular expressions representing each protocol 12 and 22. Then the interface generator 118 determines 206 the permitted sequence of operations and resolves 208 all non-determinisms. Each of these steps is described in detail below.